

①

## Classifiers algorithms :-

① - K-nearest neighbor.  
(nonparametric Statistics)

لا يعتمد على Probability

② - Bayes Decision Theory

يعتمد على Probability

لغرف وجود 2 classes  $\leftarrow w_1, w_2$  ، المراد معرفة  $(x)$  تنتمي لأي classes  
-  $P(w_i) = \frac{N_i}{N}$  ،  $i=1,2$  ، تسمى Prior

$(N_i)$  : عدد العناصر الموجودة في class رقم  $(i)$   
 $(N)$  : عدد كل العناصر الموجودة في classes

-  $P(x|w_i)$  ، تسمى Likelihood  
The class - conditional probability density function (PDF)

# القانون هو وضح بمثال (نزل مع الحاضره)

$$P(w_i|x) = \frac{P(x|w_i) \cdot P(w_i)}{P(x)} \quad i=1,2$$

-  $P(w_i|x)$  : احتمالية أن  $(x)$  تنتمي لـ  $w_i$  ، تسمى Posterior  
- هنالك نفس القانون في  $(i=1,2)$   
-  $P(x)$  ثابتة لكل مره ، يكتب في كذا في decision



$$\text{if } P(w_1|x) > P(w_2|x)$$

$$\therefore x \in w_1$$

$$\text{if } P(w_2|x) > P(w_1|x)$$

$$\therefore x \in w_2$$

① - في حالة uniform Priors  $P(w_1) = P(w_2)$

② -  $P(x)$  مشترك وبالتالي في ال decision

يكون المقارنة في خلال فقط اذا كانت  $P(w_1) = P(w_2)$

$$P(x|w_1) \geq P(x|w_2)$$

# في مثال بيوضح تطبيق ال algorithm مفصل  
عن الحاضرة في ال slides

## Feature extraction algorithms

① PCA Principal Component analysis (PCA)

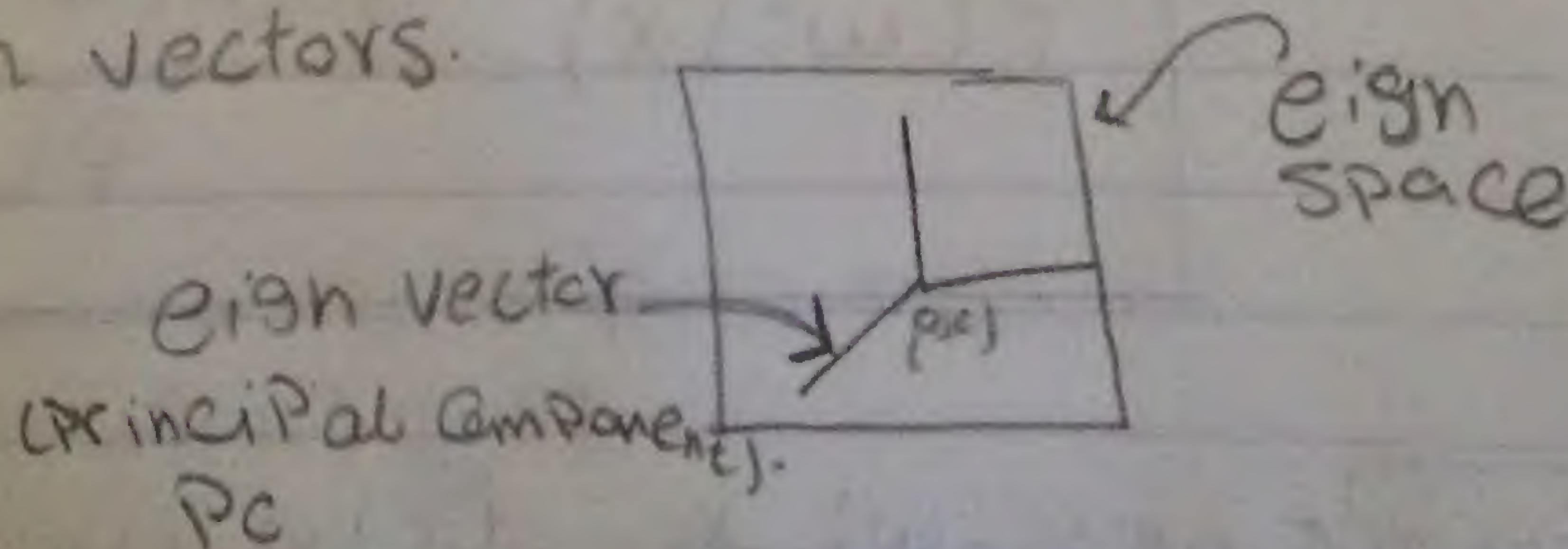
② Linear Discriminal Analysis (LDA)

③ Scale Invariant Feature Transform (SIFT)

PCA (Spatial domain Projection Eigen Domain)

② \* هيسpace domain آني لتسهيل المقارنة operatione  
(eigen space)

The Component of eigen space is  
Principal Component (PC), consists  
of eigen vectors.



② \* هيسpace reduction ← vector selection  
reduction



\* The Implementation of the algorithm are:-

1] تمثيل الصور (objects) كلها في ال data Base ك Row vector أو column vector

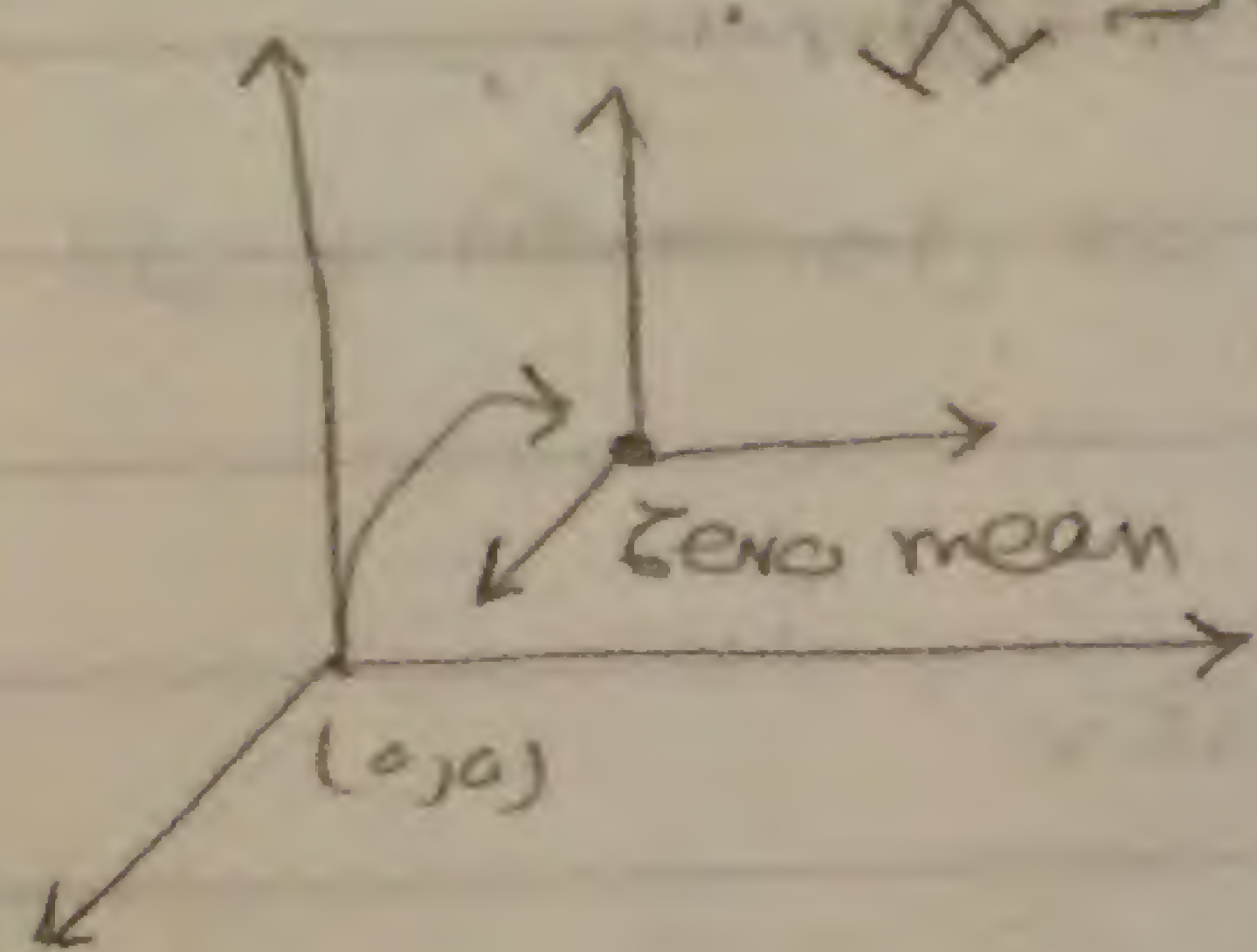
Data Base (Image) vector  $X = \begin{bmatrix} \square & \square & \square & \dots \end{bmatrix}$

$$X = [x_1, x_2, x_3, \dots]^T$$

2] حساب ال mean والمخرج من ال  $X$

فيعمل ال data لا zero mean

$$x' = x - \text{mean}$$



3] حساب ال Covariance matrix

شرح معنى ال Covariance

العلاقة بين 2 Random values  
لو ال random value  $\times$  random value Covariance

$$\text{if } (\text{Covariance}) > 0 \quad (+)$$

(علاقة طردية) فناه ان ال 2 values يزيدو مع بعض

$$\text{if } (\text{Covariance}) < 0 \quad (-)$$

(علاقة عكسية) فناه لو واحد يزيد الثاني يقل والعكس

$$\text{if } (\text{Covariance}) = 0$$

فناه ان Two Random value فاهم علاقة بعض

$$\Sigma = \sum_N (x_i - \text{mean})(x_i - \text{mean})^T$$

$$\text{mean} = (1/N) \sum_N x_i$$



4) بعد كده هيقسمه ال Covariance Matrix ال eigen vector

وبعد به هيرتب ال eigen values من الكبير للصغير و بعد Selection لأول 7 هتبقى  
وده بيتمثل ال (Principle Component) لأنها تحتوي على أكبر كمية من  
المعلومات من القيم الأخرى .  
 $K = 7$

- ال Selection بيحدد عدد ال classes

← لو كنت class 100 مثلاً ← فدا أول 100 في eigen vectors  
 $K = 100$

# Transformation matrix  $W (K \times 1)$

ديه إلى ال تحويل من spatial domain إلى eigen domain

eigen space ←  $Y = (W^T \cdot X)$  → Image vector

[eigenvectors of  $S$  ←  
corresponding to  $K$  largest eigen values

# بعد ما انتقل الصورة إلى eigen space بيدى نعمل مقارنة وانسوف  
هنتي لأي class بأستخدام classifier .

# Example - slide 40 → 44